

# Physical computing vanuit Scratch met gebruik van scratchClient – **Gevorderden**

*Bestuur servo's, LEDs en meer vanuit Scratch  
met gebruik van RPi, Arduino, scratchClient*

Hans de Jong & Gerhard Hepp

Workshop voor de Pi And More en Maker Fair  
conferenties

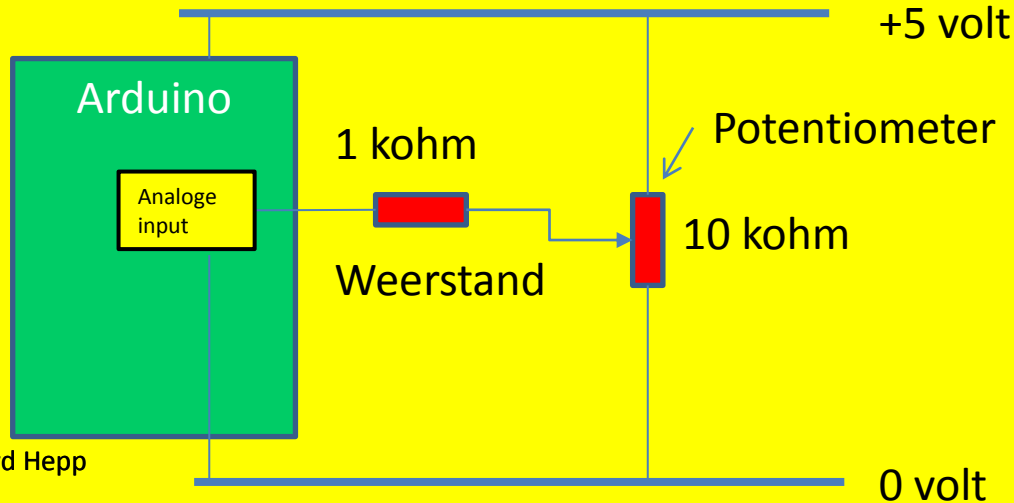
# Deel 1: Voeg analoge input toe

# Verander de bedrading op het breadboard

- Als je de bedrading gaat veranderen
  - Trek de USB kabel eruit
  - Trek de 9V adapter kabel eruit

# De elektronica van een potentiometer

- Een potmeter is een weerstand die we aansluiten tussen 5 V en 0 V.
- Over de lengte van de weerstand valt de spanning gelijkmatig (lineair).
  - Bijvoorbeeld, in het midden is het 2.5 volt
  - Dit is daarmee een analoog signaal. De waarde kan worden ingesteld tussen 5 volt en 0 volt.
- Er zit een 1 kohm weerstand in serie wegens de redenen die we eerder bespraken:

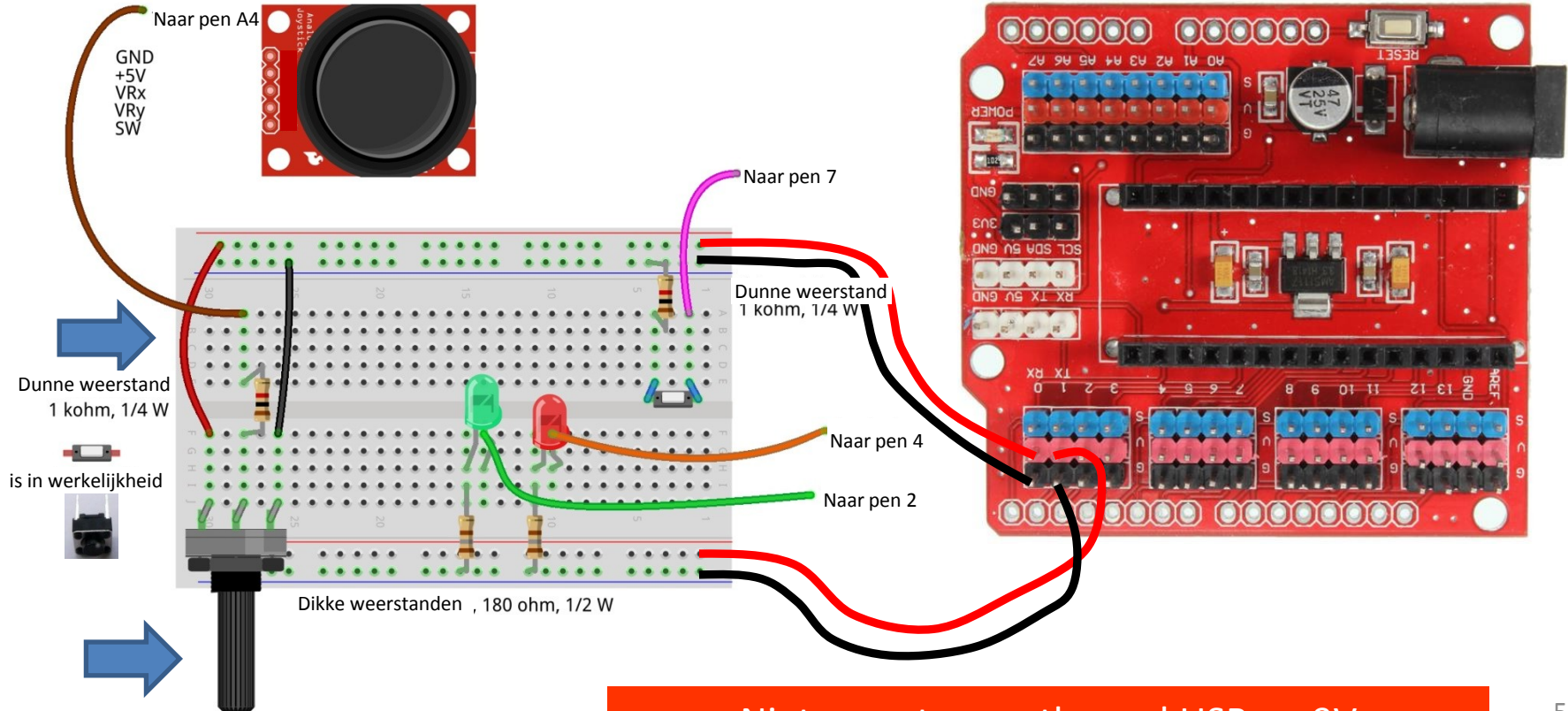


Bedenk wat er zou gebeuren als de 1 kohm weerstand er niet zat en de pen zou worden gebruikt als output en de potmeter zou dicht bij de 0 volt kant staan.

Dan zou de Arduino de pen 5 volt maken en de potmeter die spanning direct doorverbinden met de 0 volt draad → kortsluiting!

Vergelijkbaar: als de potentiometer dicht bij de 5 volt kant staat en de Arduino zou 0 volt uitgeven.

# Voeg analoge input toe



Niet vergeten: ontkoppel USB en 9V

# Werk de config file bij

- Werk de config file nogmaals bij (klik rechts op *scratchClientTutorial.scl* op het bureaublad, scratchConfig Edit)
  - Definieer *Potmeter* op pen A4 (direction: in, function: analog)
- Nu dat je toch bezig bent, doe dit alvast voor de volgende stappen:
  - Definieer *Servo1* op pen 12 (direction: out, function: servo)
  - Definieer *BigBlueLED* op pen 5 (direction: out, function: PWM)
  - Definieer *Buzzer* op pen 11 (direction: out, function: PWM)
- Vergeen niet om op te slaan. Houd de config editor open.

# Test of het werkt

- Sluit USB en 9 volt weer aan
- Stop and herstart scratchClient door dubbelklikken op *scratchClientTutorial.scl* op het bureaublad
- Maak dit stukje programma en laat het lopen
- Draai aan de knop en zie de waarden van *Potmeter* veranderen
  - Tussen ongeveer 0 and ongeveer 1023
- We schrijven in de volgende stap een stukje programma om het te gebruiken



# Deel 2: Pulsbreedte Modulatie (Pulse Width Modulation, PWM)

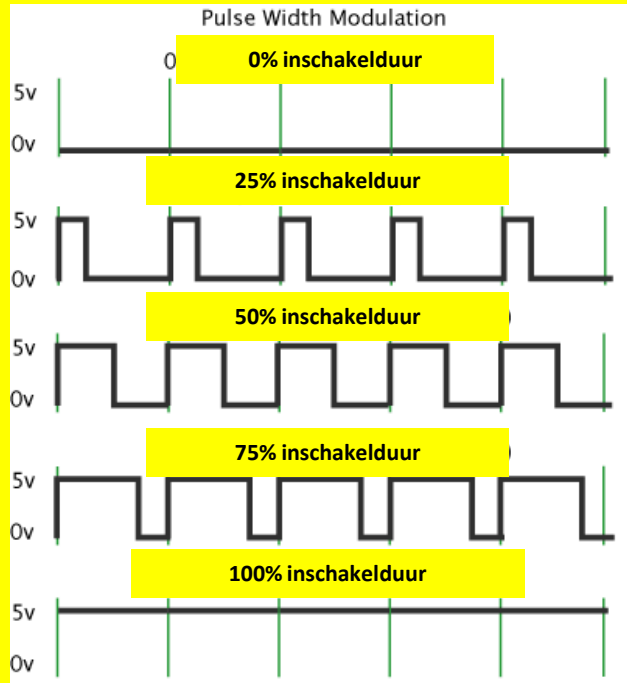


# Waar we zijn ...

- We hebben gezien:
  - Digitale Input (de knop) (in de beginnersles)
  - Digitale Output (de LED's) (in de beginnersles)
  - Analoge Input (de potentiometer)
- Dus wat is het meest logische vervolg?
  - Analoge Output? → maar dat bestaat niet!
  - Echter, er is een goed alternatief: pulsbreedte modulatie (Pulse Width Modulation, PWM)

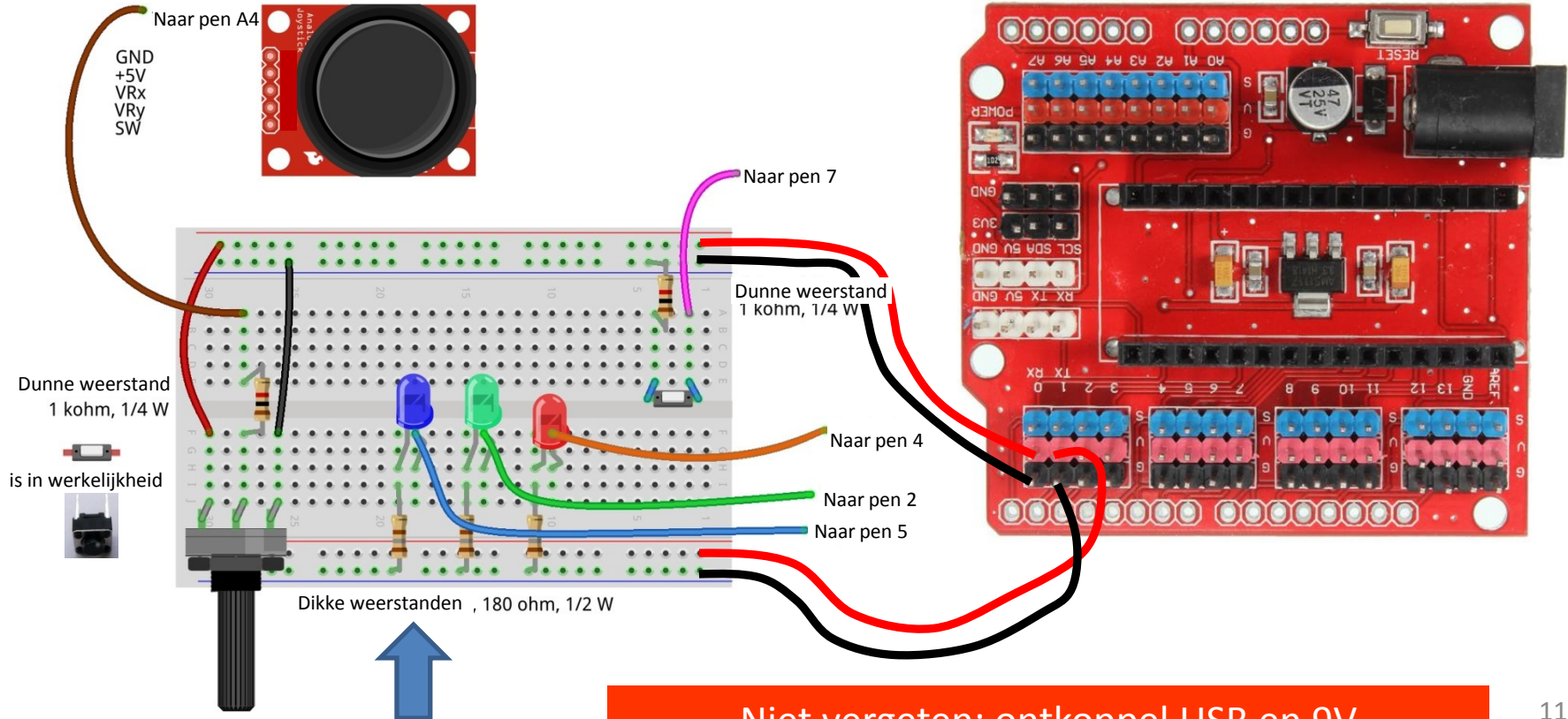
# Pulsbreedte modulatie –

## (Pulse Width Modulation, PWM)



- Door het moduleren (wijzigen) van de pulsbreedte kun je de hoeveelheid energie wijzigen die aan b.v. een LED wordt toegevoerd en daarmee de intensiteit waarmee die oplicht.
- Bedenk dat in de praktijk de LED ca. 800 maal per seconde knippert.
- Echter, niemand kan meer zien dan 100 knipperingen per seconde.

# Dim de BigBlueLED met PWM



# Test

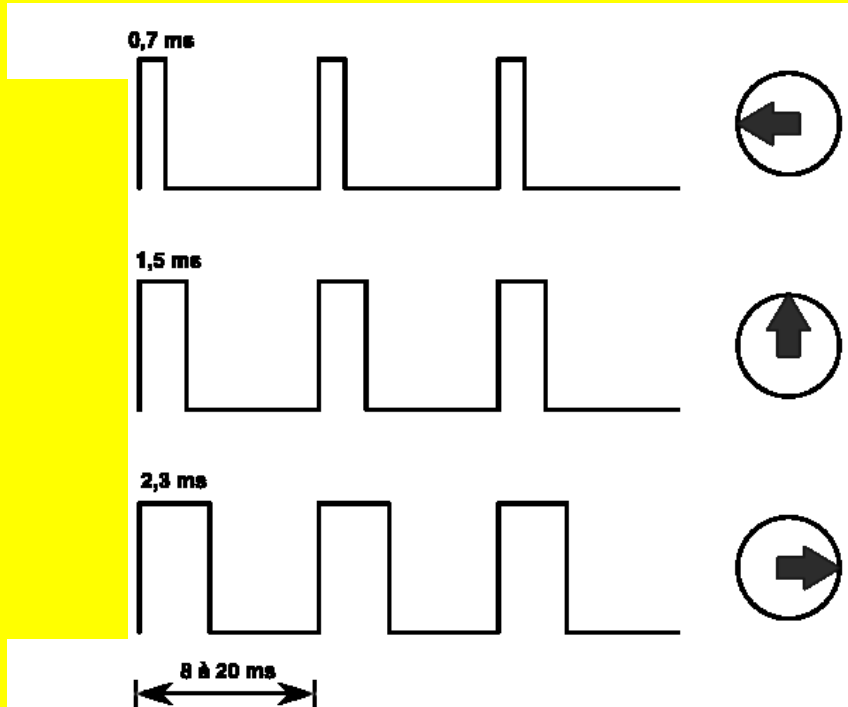
- Sluit de USB en de 9V kabel weer aan
- Je zult zien dat scratchClient de Arduino weer vindt
- De config file heb je eerder al bijgewerkt, dus het is niet nodig om scratchClient opnieuw te starten
- Zend *BigBlueLED* waarden tussen 0 en 255
  - Verandert de helderheid van de blauwe LED?

# Verbind de Potmeter en Big Blue LED (via Scratch)



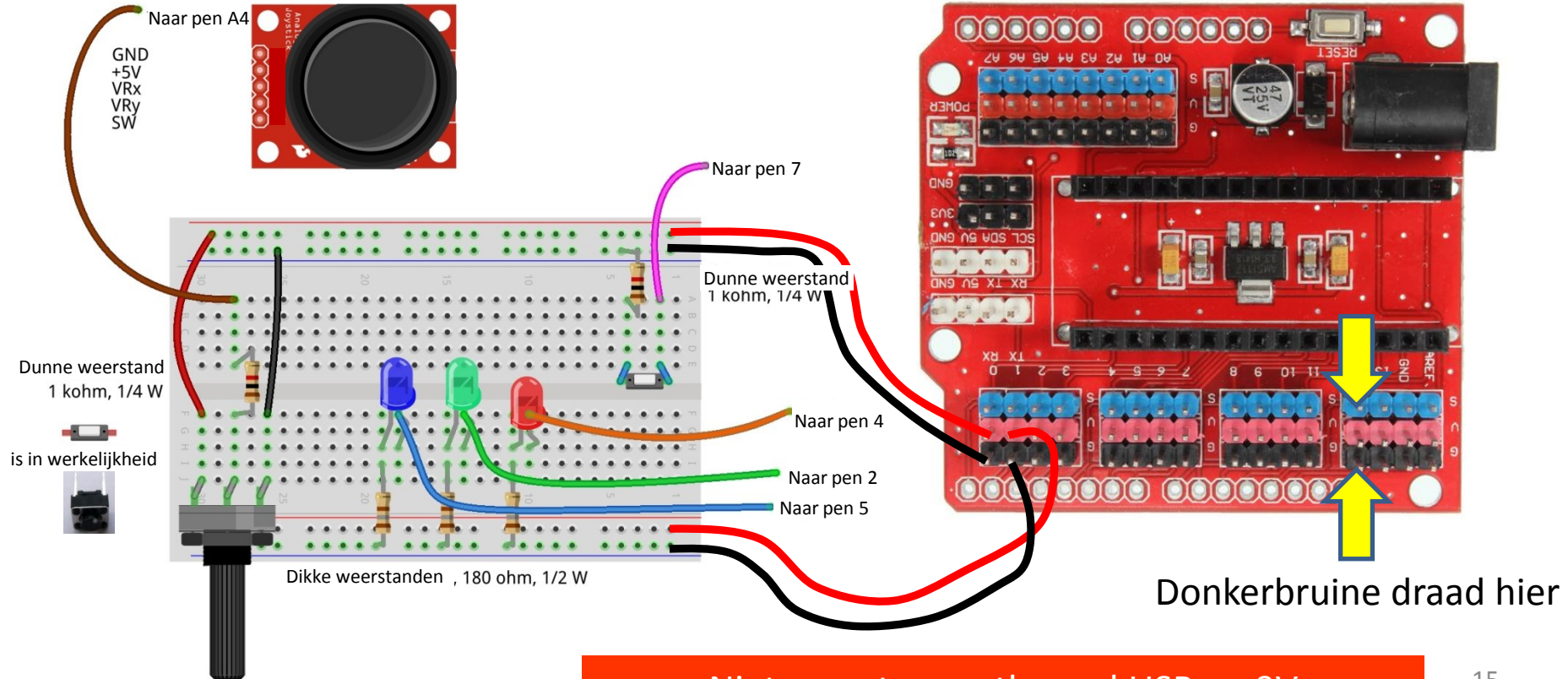
- Maak wat code die de waarde van de potentiometer (tussen 0 en 1023) omzet naar een waarde in het bereik van 0 tot 255 (deel het dus door 4) en de waarde van *BigBlueLED* zet.
- Probeer om de met de potmeter de lichtintensiteit te veranderen.
- Merk op dat als er naar rechts gedraaid wordt, de intensiteit omlaag gaat.
  - Je zou misschien verwachten dat die omhoog gaat ...
  - Hoe kun je dit simpel veranderen door twee draden om te draaien?

# Het besturen van een servo met PWM



- De positie van de servo wordt veranderd door er pulsen van verschillende breedte naar toe te sturen.
- De servo kijkt naar de pulsbreedte en draait zoals gewenst.
- De servo krijgt apart spanning.
- Bij een servo bestuurt de pulsbreedte *niet* de hoeveelheid energie die naar de servo wordt gestuurd.
- Met een servo is pulsbreedte modulatie eigenlijk een communicatieprotocol.

# Voeg de servo toe op pen 12



# Het testen van de servo



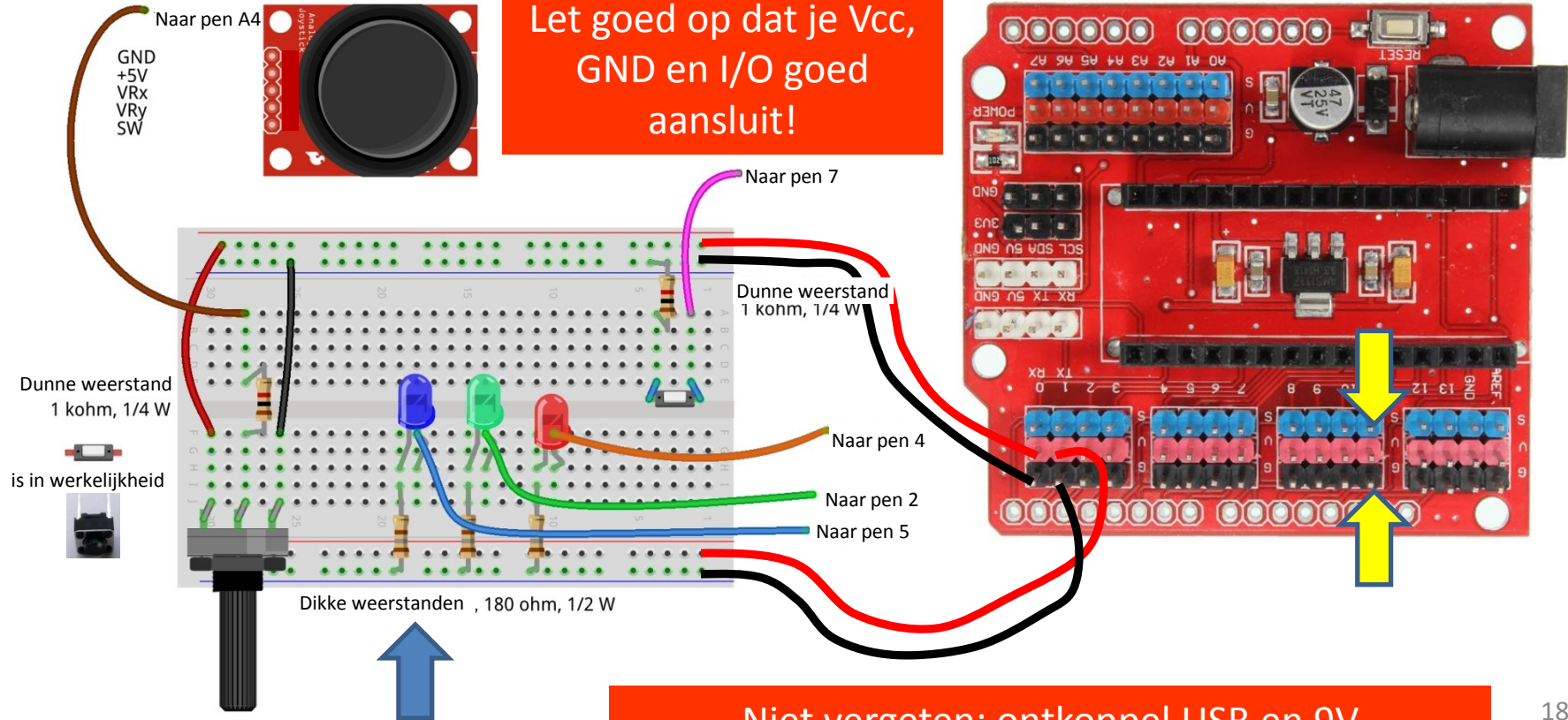
- Sluit de USB en 9V kabel aan.
- Werk de lus bij waar je de waarde van *Big Blue LED* zet om nu ook *Servo 1* een waarde te geven afhankelijk van de stand van de potmeter
  - Maar kijk uit: de waarden moeten zijn tussen 0 en 180.
  - Om overflow te voorkomen: deel eerst door 1024, vermenigvuldig dan met 180.
- Het kan zijn dat je ziet dat de servo wat kleine bewegingen maakt (jitter).
  - Omdat de waarde van de potmeter een beetje varieert en er wat variaties in de voedingspanning zijn, is de waarde van de potmeter niet constant.
  - In de geavanceerde workshop besteden we aandacht hoe we met jitter kunnen omgaan.



# Bestuur de zoemer met PWM

- De zoemer maakt geluid als die een signaal krijgt in het hoorbare frequentiegebied.
- Een PWM signaal op Arduino geeft ca. 800 Hz.
  - Verschillend op verschillende pennen
- Je zult zien dat er niet veel invloed is met het veranderen van de inschakelduur.

# Sluit de zoemer aan op pen 11



# Testen van de zoemer

- Geen noodzaak om scratchClient te herstarten, want de config file is niet gewijzigd.
- Voeg het aansturen van de zoemer toe aan de lus waar je al de servo en de Big Blue LED aanstuurt.
  - Net zoals met de LED moeten de waarden liggen tussen 0 and 255.
- Test alleen een korte tijd (om de oren van je burens te sparen 😊).

# PWM beperkingen van Arduino (Nano en Uno)

- Algemene PWM is beschikbaar op pennen 3, 5, 6, 9, 10, 11
- De servo kan worden geconfigureerd op die pennen, maar ook op pennen 2, 4, 7, 8, 12
- Als er een servo is geconfigureerd op enige pen, dan kunnen pennen 9 en 10 niet meer als PWM geconfigureerd worden.
  - De config tool zal waarschuwen als je het fout doet.

# Deel 3: Samenvatting en punten om te onthouden

# Onthouden na de intermediate workshop

- Met scratchClient kun je definiëren:
  - De functie van elke pen
  - Een symbolische naam voor elke pen
- scratchClient config is het programma om de configuratie te doen
- Herstart scratchClient nadat je de configuratie hebt veranderd
- Zet een weerstand in serie met de LED's
- Zet een weerstand in serie met de schakelaars
- Configureer een optrekweerstand (pull up resistor) als het signaal varieert tussen 0 volt en open in plaats van tussen 0 volt en 3 tot 5 volt.
- In Scratch 2, gebruik extensieblokken voor scratchClient om blokken te krijgen waarmee je met scratchClient kunt werken
- Je kunt de waarden van alle pennen in de gaten houden in de browser
- **scratchClient kan nog veel meer doen ...**
- Er zijn wat merkwaardigheden, daarom
  - Sla de Scratch 2 bestanden op in de thuismap (zoals Scratch 2 voorstelt)
  - Laat *CleanUpScratchFiles.sh* lopen. Die staat op het bureaublad. Eenmalig na eerste opstart van de RPi. Laat het dan installeren in crontab.
  - Werk vanaf het bureaublad
  - Klik op het groene vierkantje elke keer nadat je een project met scratchClient heropent.
- Een pen op Arduino kan deze functies hebben:
  - Digitaal In
  - Digitaal Uit
  - Analooog In
  - *Geen Analooog Uit*
  - Pulsbreedte modulatie (Pulse Width Modulation, PWM) als alternatief voor het ontbreken van Analooog Uit
    - Voor het sturen van de intensiteit van een LED
    - Voor het besturen van een servo
    - Voor het besturen van een zoemer
  - Er zijn een paar meer, zie het geavanceerde niveau
  - Je kunt optrekweerstanden (pull up resistors) configureren op de Digitale In

Eind van de workshop voor  
**gevorderden**